VLSI Memo No. 89-500
January 1989

The Organization of Permutation Architectures with Bussed Interconnections

Joe Kilian, Shlomo Kipnis, Charles E. Leiserson

Abstract

This paper explores the problem of efficiently permuting data stored in VLSI chips in accordance with a predetermined set of permutations. By connecting chips with shared bus interconnections, as opposed to point-to-point interconnections, we show that the number of pins per chip can often be reduced. For example, for infinitely many n, we exhibit permutation architectures with $\lceil \sqrt{n} \rceil$ pins per chip that can realize any of the n cyclic shifts on n chips in one clock tick. When the set of permutations forms a group with p elements, any permutation in the group can be realized in one clock tick by an architecture with $O(\sqrt{p \lg p})$ pins per chip. When the permutation group is abelian, $O(\sqrt{p})$ pins suffice. These results are all derived from a mathematical characterization of *uniform permutation architectures* based on the combinatorial notion of a *difference cover*. We also consider uniform permutation architectures that realize permutations in several clock ticks, instead of one, and show that further savings in the number of pins per chip can be obtained.

Microsystems Research Center Room 39-321
Massachusetts Institute of Technology
Cambridge Massachusetts 02139
Telephone (617) 253-8138

Acknowledgements

Appeared in the *IEEE, 28th Annual Symposium on Foundations of Computer Science,* October 12-14, 1987, pp. 305-315. This research is supported in part by the Defense Advanced Research Projects Agency under contract N00014-80-C-0622, by a Fannie and John Hertz Foundation fellowship, and by an NSF Presidential Young Investigator Award with matching funds from AT&T Bell Laboratories, IBM Corporation, and Xerox Corporation.

Author Information

Kilian: Laboratory for Computer Science, Room NE43-330, MIT, Cambridge, MA 02139. (617) 253-6259.

Kipnis: Laboratory for Computer Science, Room NE43-311, MIT, Cambridge, MA 02139. (617) 253-2345.

Leiserson: Laboratory for Computer Science, Room NE43-321, MIT, Cambridge, MA 02139. (617) 253-5833.

The Organization of Permutation Architectures with Bussed Interconnections

Joe Kilian
Shlomo Kipnis
Charles E. Leiserson

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

January 12, 1989

Abstract

This paper explores the problem of efficiently permuting data stored in VLSI chips in accordance with a predetermined set of permutations. By connecting chips with shared bus interconnections, as opposed to point-to-point interconnections, we show that the number of pins per chip can often be reduced. For example, for infinitely many n, we exhibit permutation architectures with $\lceil \sqrt{n} \rceil$ pins per chip that can realize any of the n cyclic shifts on n chips in one clock tick. When the set of permutations forms a group with p elements, any permutation in the group can be realized in one clock tick by an architecture with $O(\sqrt{p \lg p})$ pins per chip. When the permutation group is abelian, $O(\sqrt{p})$ pins suffice. These results are all derived from a mathematical characterization of *uniform permutation architectures* based on the combinatorial notion of a *difference cover*. We also consider uniform permutation architectures that realize permutations in several clock ticks, instead of one, and show that further savings in the number of pins per chip can be obtained.

Keywords: barrel shifter, bussed interconnections, cyclic shifter, difference cover, difference set, group theory, permutation, permutation architecture, projective plane, special-purpose architecture, uniform architecture.

[0]This research was supported in part by the Defense Advanced Research Projects Agency under Contract N00014-80-C-0622. Joe Kilian is supported in part by a Fannie and John Hertz Foundation fellowship. Charles Leiserson is supported in part by an NSF Presidential Young Investigator Award with matching funds provided by AT&T Bell Laboratories, IBM Corporation, and Xerox Corporation.

1 Introduction

The organization of communication among chips is a major concern in the design of an electronic system. Because of the costs associated with wiring and packaging, it is generally desirable to minimize the number of wires and the number of pins per chip in an architecture. This paper investigates how busses (multiple-pin wires) can be employed to efficiently implement various communication patterns among a set of chips. Other theoretical studies of bussed interconnections can be found in [1, 3, 4, 5, 7, 12, 21, 24, 25, 29].

Perhaps the simplest example of the advantage of bussed interconnections is the use of a single shared bus to communicate between any pair of chips connected to the bus in one clock tick. Communicating between any pair of chips in one clock tick can be implemented with two-pin wires, but any such scheme requires $\binom{n}{2}$ wires and $n-1$ pins per chip.[1] Of course, a two-pin interconnection scheme may be able to implement more communication patterns, but if we are only interested in communication between individual pairs, the additional power, which comes at a high cost, is wasted.

An example that better illustrates the ideas in this paper comes from the problem of building a fast *cyclic shifter* (sometimes called a *barrel shifter*) on n chips. Initially, each chip c contains a one-bit value ϵ_c. The function of the shifter is to move each bit ϵ_c to chip $c+s \pmod n$ in one clock tick, where s can be any value between 0 and $n-1$.

Any cyclic shifter that uses only two-pin wires requires at least $\binom{n}{2}$ wires and $n-1$ pins per chip in order to shift in one clock tick because each chip must be able to communicate directly with each of the other $n-1$ chips. Using busses, however, we can do much better. Figure 1 gives an architecture for a cyclic shifter on 13 chips which uses 13 busses and only 4 pins per chip. To realize a shift by 8, for example, each chip writes its bit to pin 3 and reads from pin 1. The reader may verify that all other cyclic shifts among the chips are possible in one clock tick. (In Section 4, we give a general method for constructing such cyclic shifters based on finite projective planes.)

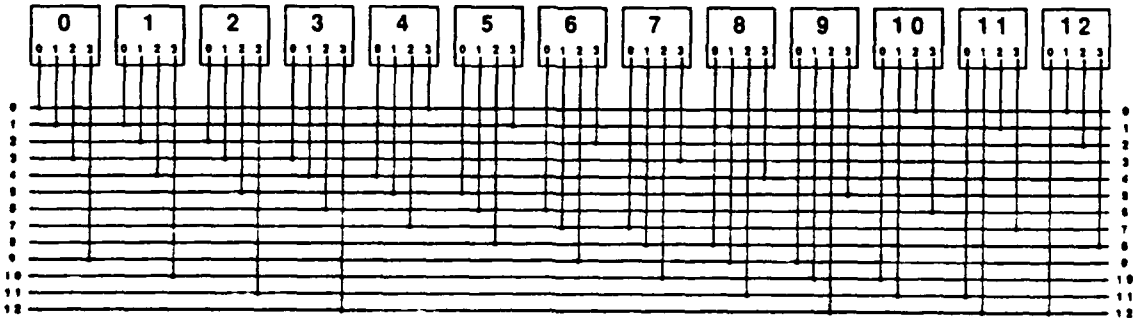


Figure 1: A cyclic shifter on 13 chips that uses 13 busses. Each chip has 4 pins, and each bus has 4 chips connected to it. This cyclic shifter is based on the difference cover $\{0,1,3,9\}$ for Z_{13}.

The cyclic shifter of Figure 1 has the advantage of uniformity. All chips have exactly the same number of pins, and to accomplish each of the 13 permutations specified by the

[1] Unless otherwise specified, we count only data pins in our analysis and omit consideration of the pins for control, clock, power, and ground since they are needed by all implementations.

problem, all chips write to (and read from) pins with identical labels. For all busses, the number of pins per bus is 4, which is the same as the number of pins per chip. Moreover, the connections between chips and busses follow a periodic pattern. The uniformity of the architecture leads to simplicity in the control of the system. Four control wires from a central controller are sufficient to determine each of the 13 shifts—two wires for specifying the number of the pin on which to write, and two for the pin to read—which is the minimum possible. Thus, our control scheme uses the minimum number of control pins, and the on-chip decoding logic is straightforward and identical for all the chips.

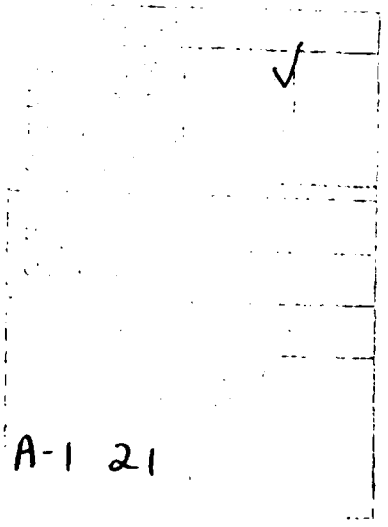
Cyclic shifters for general n can be constructed using an idea from combinatorial mathematics related to difference sets [18, p. 121]. (See also [6, 14, 16, 22, 26].)

Definition 1 A subset $D \subseteq \mathbf{Z}_n$ of the integers modulo n is a *difference cover for* $\mathbf{Z}_n$ if for all $s \in \mathbf{Z}_n$, there exist $d_i, d_j \in D$ such that $s = d_i - d_j \pmod{n}$.

That is, every integer in $\mathbf{Z}_n$ can be represented as the difference modulo n of two integers in D. For example, the set $D = \{0, 1, 3, 9\}$ is a difference cover for $\mathbf{Z}_{13}$, since

$$\begin{aligned}
0 &= 0-0 \\
1 &= 1-0 \\
2 &= 3-1 \\
3 &= 3-0 \\
4 &= 0-9 \\
5 &= 1-9 \\
6 &= 9-3 \\
7 &= 3-9 \\
8 &= 9-1 \\
9 &= 9-0 \\
10 &= 0-3 \\
11 &= 1-3 \\
12 &= 0-1\,,
\end{aligned}$$

where all subtractions are performed modulo 13.



Given a difference cover for $\mathbf{Z}_n$ with k elements, a cyclic shifter on n chips with n busses and k pins per chip can be constructed. Suppose $D = \{d_0, d_1, \ldots, d_{k-1}\}$ is a difference cover for $\mathbf{Z}_n$. In the cyclic shifter, chip c connects via its pin i to bus $c + d_i \pmod{n}$, for all $c = 0, 1, \ldots, n-1$ and $i = 0, 1, \ldots, k-1$. To see that any cyclic shift on the n chips can be uniformly realized, consider a cyclic shift by s. Since D is a difference cover for $\mathbf{Z}_n$, there exist $d_i, d_j \in D$ such that $s = d_i - d_j \pmod{n}$. To realize the shift by s, each chip writes to pin i and reads from pin j. Chip c therefore writes onto bus $c + d_i$, and bus $c + d_i$ is read by chip $(c + d_i) - d_j = c + s$. No collisions occur because each bus has exactly one pin labeled i and one pin labeled j connected to it, as can be verified.

The remainder of this paper explores permutation architectures, the properties of multiple-pin interconnections, and related combinatorial mathematics. In Section 2 we

define a permutation architecture, introduce the notion of uniformity, and prove some basic properties of architectures that employ busses to realize arbitrary sets of permutations. Section 3 defines the notion of a difference cover for a set of permutations, relates it to the notion of a uniform permutation architecture, and proves some properties of difference covers. In Section 4 we show how to build cyclic shifters that are provably efficient. Section 5 investigates how to design small difference covers for any set of permutations that forms a finite group. In Section 6 we extend the discussion to uniform architectures that realize permutations in more than one clock tick. We present a variety of extensions to the results of the paper in Section 7. Finally, in Section 8 we discuss questions left open by our research. An appendix of standard notations and definitions is included for reference. Notations and definitions more specific to the content of the paper are provided in context.

2 Permutation architectures

In this section we formally define the notion of a permutation architecture, and we make precise the notion of uniformity. We also prove some basic properties of permutation architectures that realize arbitrary sets of permutations. The definitions in this section are somewhat intricate and tedious, and are indicative of the difficulties faced in the design of efficient permutation architectures. In the next section, however, we use these definitions to show that reasoning about uniform permutation architectures is essentially equivalent to reasoning about difference covers, a simpler and more elegant mathematical notion. The remainder of the paper then uses the simpler notion.

For convenience, we adopt a few notational conventions. We use multiplicative notation to denote composition of permutations. The inverse of a permutation π is denoted by π^{-1}. Composition of functions is performed in right-to-left order, so that $\pi_1\pi_2$ is defined by $\pi_1\pi_2 x = \pi_1(\pi_2(x))$. The identity permutation on n elements is denoted by I_n, or by I if the number of elements is unimportant. For a permutation set Φ, we denote by Φ^{-1} the set of all the inverses of the permutations of Φ, i.e. $\Phi^{-1} = \{\phi^{-1} : \phi \in \Phi\}$. For two permutation sets Φ and Ψ, the notation $\Phi\Psi$ is used to denote the permutation set $\{\phi\psi : \phi \in \Phi \text{ and } \psi \in \Psi\}$. We use the notation $[n]$ to denote the set of n integers $\{0, 1, \ldots, n-1\}$.

We first define the notion of a permutation architecture.

Definition 2 A *permutation architecture* is a 6-tuple $\mathcal{A} = (C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL})$ as follows.

1. C is a set of *chips*;
2. B is a set of *busses*;
3. P is a set of *pins*;
4. CHIP is a function $\text{CHIP} : P \to C$;

5. BUS is a function $\text{BUS} : P \rightarrow B$;

6. LABEL is a function $\text{LABEL} : P \rightarrow \mathbf{N}$, where if $x, y \in P$, $x \neq y$, and $\text{CHIP}(x) = \text{CHIP}(y)$, then $\text{LABEL}(x) \neq \text{LABEL}(y)$.

The set C contains all the chips in the architecture, and the set B contains all the busses. Which chips are connected to which busses is determined by the pins they have in common; the set P contains all the pins. The function CHIP determines which pins belong to which chips. Similarly, the function BUS determines which pins are interconnected by which bus. The function LABEL names the pins on the chips by natural numbers such that all pins on a given chip have distinct labels, which we shall sometimes call pin numbers.

Our formal definition of a permutation architecture omits several subsystems that technically should be included, but whose inclusion is not germane to our study. These subsystems include a control network that specifies what permutation is to be performed and clocking circuitry for synchronization. Our focus is on the structure of the bussed interconnections for permuting the data, and thus our definition encompasses only this aspect of the architecture.

We now define what it means for a permutation architecture to realize a permutation.

Definition 3 A permutation architecture $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL} \rangle$ *realizes* a permutation $\pi : C \rightarrow C$ if there exist two functions $\text{WRITE}_\pi : C \rightarrow P$ and $\text{READ}_\pi : C \rightarrow P$, such that for any chips $c, c_1, c_2 \in C$, we have:

1. $\text{CHIP}(\text{READ}_\pi(c)) = \text{CHIP}(\text{WRITE}_\pi(c)) = c$;

2. $\text{BUS}(\text{WRITE}_\pi(c)) = \text{BUS}(\text{READ}_\pi(\pi(c)))$;

3. $c_1 \neq c_2$ implies $\text{BUS}(\text{WRITE}_\pi(c_1)) \neq \text{BUS}(\text{WRITE}_\pi(c_2))$.

The architecture *uniformly realizes* π if, in addition:

4. $\text{LABEL}(\text{WRITE}_\pi(c_1)) = \text{LABEL}(\text{WRITE}_\pi(c_2))$;

5. $\text{LABEL}(\text{READ}_\pi(c_1)) = \text{LABEL}(\text{READ}_\pi(c_2))$.

We say a permutation architecture *realizes* a set Π of permutations if it realizes every permutation in Π. We say it *uniformly realizes* Π if it uniformly realizes every permutation in Π.

Intuitively, for a permutation π, the functions WRITE_π and READ_π identify the *write pin* and the *read pin* for each chip. Condition 1 makes sure that each chip writes and reads pins that are connected to it. Condition 2 ensures that the bus to which chip c writes is read by chip $\pi(c)$. Condition 3 guarantees that no collisions occur, that is, no two data transfers use the same bus. The architecture uniformly realizes a permutation (Conditions 4 and 5) if all chips write to pins with the same pin number and read from pins with the same pin number, as in the cyclic shifter from Figure 1.

Our definition of a permutation architecture implies that "complete" permutations are to be realized, that is, every chip sends exactly one datum and receives exactly one datum. Moreover, an interconnection is required even when a chip sends a datum to itself. Since no collisions occur, the number of busses in the architecture must be at least the number of chips. This observation leads directly to the following theorem.

Theorem 1 *In any permutation architecture that realizes some nonempty permutation set* Π*, the average number of pins per bus is at most the average number of pins per chip.*

Proof. Let $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL} \rangle$ be a permutation architecture for Π. The average number of pins per chip is $|P| / |C|$, and the average number of pins per bus is $|P| / |B|$. Condition 3 of Definition 3 says that for any permutation $\pi \in \Pi$, any two distinct chips are mapped to distinct busses. Consequently, we get that $|B| \geq |C|$, which proves the theorem. ■

Under the assumption that no interconnection is needed for a chip to send data to itself, Theorem 1 is no longer applicable. A similar theorem can be proved for this model, however, which involves the number of fixed points in the permutations realized by the architecture. Specifically, suppose the architecture realizes a set Π of permutations. Define the *rank* of a permutation $\pi \in \Pi$ as $\text{RANK}(\pi) = |\{c \in C : \pi(c) \neq c\}|$, and define the rank of the permutation set Π as $\text{RANK}(\Pi) = \max_{\pi \in \Pi} \text{RANK}(\pi)$. The analogue to Theorem 1 states that the ratio between the average number of pins per bus and the average number of pins per chip is at most $|C| / \text{RANK}(\Pi)$.

In any architecture $\mathcal{A}$ that uniformly realizes a permutation set Π, the number of pins that are actually used to uniformly realize Π is the same for all chips, and additional pins on a chip are unused. Furthermore, the number of busses used in realizing any permutation $\pi \in \Pi$ is equal to the number of chips. These observations lead to the following definition of a uniform architecture.

Definition 4 A *uniform permutation architecture* for a permutation set Π is a permutation architecture $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL} \rangle$ such that:

1. $\mathcal{A}$ uniformly realizes Π;
2. $|\{x \in P : \text{CHIP}(p) = c_1\}| = |\{x \in P : \text{CHIP}(p) = c_2\}|$ for any two chips $c_1, c_2 \in C$;
3. $|B| = |C|$;
4. if $x \neq y$ and $\text{LABEL}(x) = \text{LABEL}(y)$, then $\text{BUS}(x) \neq \text{BUS}(y)$.

Thus, all the chips in a uniform permutation architecture have the same number of pins (Condition 2), the number of busses is equal to the number of chips (Condition 3), and the labels of the pins on any bus are distinct (Condition 4).

The following theorem demonstrates that any permutation architecture that uniformly realizes some permutation set Π can be made into a uniform architecture.

Theorem 2 *Let $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL}\rangle$ be a permutation architecture that uniformly realizes the permutation set Π, and let k be the smallest number of pins on any chip in C. Then there is a uniform architecture $\mathcal{A}' = \langle C', B', P', \text{CHIP}', \text{BUS}', \text{LABEL}'\rangle$ for Π with at most k pins per chip.*

Proof. We construct the uniform architecture $\mathcal{A}'$ from the permutation architecture $\mathcal{A}$ in two steps. First, we construct an intermediate permutation architecture $\mathcal{A}'' = \langle C'', B'', P'', \text{CHIP}'', \text{BUS}'', \text{LABEL}''\rangle$ by removing extraneous pins from chips in $\mathcal{A}$ such that all chips end up with the same number of pins per chip and such that each pin plays a role in uniformly realizing Π. Then, the busses of $\mathcal{A}''$ are reorganized to produce the architecture $\mathcal{A}'$ in such a way that the number of busses in $\mathcal{A}'$ is equal to the number of chips. We assume that the permutation set Π is nonempty, since otherwise the theorem is trivial.

In the first step, we remove pins that are unused in uniformly realizing Π. Since $\mathcal{A}$ uniformly realizes Π, each permutation $\pi \in \Pi$ can be associated with a distinct pair (i, j) of pin labels corresponding to the labels that all chips write to and read from in order to realize π. A pin is unused if its label does not appear in any of these $|\Pi|$ pairs. Removing the unused pins results in the architecture $\mathcal{A}''$ in which all chips have the same number of pins, since each chip has exactly one pin for each label used in uniformly realizing Π. The permutation architecture $\mathcal{A}''$ uniformly realizes Π, and furthermore, each pin is used in uniformly realizing some $\pi \in \Pi$. If we let s denote the number of pins per chip in $\mathcal{A}''$, then we have $s \leq k$, since originally at least one chip had k pins and no pins were added.

In the second step, we reorganize the busses of $\mathcal{A}''$ to produce the uniform architecture $\mathcal{A}'$ in which the number of busses is equal to the number of chips. For any permutation architecture that realizes a nonempty permutation set, the number of busses is never smaller that the number of chips. Assume without loss of generality that $C'' = [n]$, $B'' = [m]$, and range(LABEL$''$) $= [s]$. The theorem is proved if the architecture $\mathcal{A}''$ uses only $n = |C''|$ busses, but in general, the architecture might use $m > n$ busses.

We define a collection of mappings $\Psi = \{\psi_0, \psi_1, \ldots, \psi_{s-1}\}$, where for each $0 \leq i \leq s-1$, the mapping $\psi_i : [n] \rightarrow [m]$ is defined to be $\psi_i(c) = b$ if and only if chip $c \in C''$ is connected via its pin number i to bus $b \in B''$. The elements of Ψ are indeed mappings since each chip has a pin numbered i for each $0 \leq i \leq s-1$. The mappings are injective (one-to-one), since otherwise two pins with the same pin number would be connected to the same bus, and both pins could not be used to uniformly realize permutations, thereby violating the construction of $\mathcal{A}''$ in the first step. The collection Ψ is a multiset, since it may be that two different pin numbers $i \neq j$ define the same mapping (i.e. $\psi_i = \psi_j$). The key idea is that any permutation is implemented by each chip writing to pin i and reading from pin j, thereby employing the mapping ψ_i to write data from the n chips to n distinct busses, and the inverse of the mapping ψ_j to read data from the same n busses back to the n chips.

We now show how to reorganize the busses of $\mathcal{A}''$ in order to construct a uniform architecture $\mathcal{A}'$. We partition Ψ into l equivalence classes $\Psi_0 \cup \Psi_1 \cup \cdots \cup \Psi_{l-1}$ such that ψ_i and ψ_j are in the same equivalence class Ψ_r, if and only if range(ψ_i) = range(ψ_j). This partitioning has the property that if $\pi \in \Pi$, then there exists an r such that $\pi = \psi_j^{-1}\psi_i$,

where $\psi_i, \psi_j \in \Psi_r$. (Recall that the inverse of an injective mapping $\psi : [n] \rightarrow [m]$ is defined as the mapping $\psi^{-1} : \text{range}(\psi) \rightarrow [n]$ such that if $\psi(c) = b$, then $\psi^{-1}(b) = c$.) For each $0 \leq r \leq l-1$, pick a bijection (one-to-one, onto) $f_r : \text{range}(\psi) \rightarrow [n]$, where ψ is any mapping in Ψ_r. (We can pick a bijection, since ψ is injective, which implies $|\text{range}(\psi)| = n$.) We define the architecture $\mathcal{A}'$ by $C' = C''$, $B' = [n]$, $P' = P''$, CHIP$'$ = CHIP$''$, LABEL$'$ = LABEL$''$, and for any pin $x \in P'$ such that $\psi_{\text{LABEL}'(x)} \in \Psi_r$, we define BUS$'(x) = f_r(\text{BUS}''(x))$.

The architecture $\mathcal{A}'$ has exactly s pins per chip and satisfies $|B'| = |C'| = n$, thereby satisfying Conditions 2 and 3 of Definition 4. We show Condition 4 holds by considering any two pins x and y with LABEL$'(x) =$ LABEL$'(y) = i$. We have BUS$'(x) = f_r(\text{BUS}''(x))$ and BUS$'(y) = f_r(\text{BUS}''(y))$ for some f_r as defined in the previous paragraph. Since f_r is an injective mapping and because Condition 4 of Definition 4 holds for $\mathcal{A}''$, we then have $x \neq y$ implies BUS$'(x) \neq$ BUS$'(y)$.

It remains to show that Condition 1 of Definition 4 holds, that is, that $\mathcal{A}'$ uniformly realizes Π. Consider any permutation $\pi \in \Pi$. Since $\mathcal{A}''$ uniformly realizes Π, there exists a pair of pin labels (i, j) such that π is realized in $\mathcal{A}''$ by each chip writing to its pin numbered i and reading from its pin numbered j. We use the same pin labels (i, j) to realize the permutation π in $\mathcal{A}'$. Conditions 1, 4, and 5 of Definition 3 are immediately satisfied. To verify Conditions 2 and 3 we use the following observation. In architecture $\mathcal{A}''$ chip c is connected via its pin labeled h to bus $\psi_h(c)$, while in architecture $\mathcal{A}'$ it is connected to bus $f_r(\psi_h(c))$, where $\psi_h \in \Psi_r$. Condition 2 now holds since $\pi = \psi_j^{-1}\psi_i = (f_r\psi_j)^{-1}(f_r\psi_i)$. Condition 3 holds since $f_r\psi_i$ is a permutation on $[n]$. We therefore conclude that $\mathcal{A}''$ is a uniform architecture for Π with at most k pins per chip. ■

The next theorem provides a lower bound on the number of pins per chip in any uniform architecture for a permutation set Π. (A related theorem due to C. Fiduccia appears in [20, p. 308].)

Theorem 3 *Let $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL}\rangle$ be a uniform permutation architecture for a permutation set Π. Then the number of pins per chip in $\mathcal{A}$ is at least $\sqrt{|\Pi|}$.*

Proof. Because architecture $\mathcal{A}$ realizes Π uniformly, we can associate each $\pi \in \Pi$ with a pair (i, j) of pin numbers such that π is realized by each chip writing to its pin labeled i and reading from its pin labeled j. Since $\mathcal{A}$ is uniform, each chip has exactly $|P| / |C|$ pins, and the number of such pairs is $(|P| / |C|)^2$. No two permutations can be associated with the same pair, and thus, we have $(|P| / |C|)^2 \geq |\Pi|$ or $|P| / |C| \geq \sqrt{|\Pi|}$. ■

A permutation architecture can often nonuniformly realize many more permutations than the square of the number of pins per chip. As an example, consider a "crossbar" architecture of n chips and n busses where each chip is connected to each bus. This architecture can nonuniformly realize all $n!$ permutations, which is much greater than n^2, the square of the number of pins per chip. In Section 7 we discuss some of the capabilities of nonuniform permutation architectures.

3 Difference covers

In this section, we present our main theorems which establish the relationship between difference covers for permutation sets and uniform permutation architectures. We also prove some lemmas concerning difference covers for Cartesian products of permutation sets. Finally, we present an alternative representation for difference covers called substring covers based on similar notions in the literature of difference sets.

We first provide a generalization of Definition 1 to arbitrary sets of permutations.

Definition 5 A *difference cover* for a permutation set Π is a set $\Phi = \{\phi_0, \phi_1, \ldots, \phi_{k-1}\}$ of permutations such that for each $\pi \in \Pi$ there exist $\phi_i, \phi_j \in \Phi$ such that $\pi = \phi_j^{-1}\phi_i$.

Equivalently, we can use our product-of-sets notation to say that Φ is a difference cover for Π if $\Phi^{-1}\Phi \supseteq \Pi$.

The following two theorems show how difference covers and uniform architectures are related. Theorem 4 describes how to design a uniform architecture for a permutation set Π when a difference cover for Π is given. Theorem 5 presents a construction of a difference cover for a permutation set Π from a uniform architecture for Π.

Theorem 4 *Let Π be a permutation set, and let Φ be a difference cover for Π such that $|\Phi| = k$. Then there exists a uniform architecture for Π with k pins per chip.*

Proof. Let $\Phi = \{\phi_0, \phi_1, \ldots, \phi_{k-1}\}$, and assume that Π is a set of permutations on n objects. We construct a permutation architecture for Π with n busses and k pins per chip. We name the chips and busses of the architecture by natural numbers, and the pins by pairs of natural numbers. The architecture $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL}\rangle$ is defined as $C = [n]$, $B = [n]$, $P = [n] \times [k]$, $\text{CHIP}(c, i) = c$, $\text{LABEL}(c, i) = i$, and $\text{BUS}(c, i) = \phi_{\text{LABEL}(c,i)}(\text{CHIP}(c, i)) = \phi_i(c)$. That is, chip c is connected via its pin number i to bus $\phi_i(c)$.

To see formally that this architecture uniformly realizes Π, let $\pi \in \Pi$ be a permutation, and let $\phi_i, \phi_j \in \Phi$ be elements of the difference cover for Π such that $\pi = \phi_j^{-1}\phi_i$. Define the write function for π as $\text{WRITE}_\pi(c) = (c, i)$ and define the read function for π as $\text{READ}_\pi(c) = (c, j)$. (Note that i and j are always in the range 0 through $k - 1$.) We now verify that the five Conditions of Definition 3 are satisfied. Condition 1 holds since for any chip $c \in C$ we have $\text{CHIP}(\text{WRITE}_\pi(c)) = \text{CHIP}(c, i) = c$, and $\text{CHIP}(\text{READ}_\pi(c)) = \text{CHIP}(c, j) = c$. Condition 2 is satisfied since for any chip $c \in C$ we have

$$\begin{aligned}
\text{BUS}(\text{WRITE}_\pi(c)) &= \text{BUS}(c, i) \\
&= \phi_i(c) \\
&= \phi_j\phi_j^{-1}\phi_i(c) \\
&= \phi_j(\pi(c)) \\
&= \text{BUS}(\pi(c), j) \\
&= \text{BUS}(\text{READ}_\pi(\pi(c))).
\end{aligned}$$

Condition 3 holds because if $\text{BUS}(\text{WRITE}_\pi(c_1)) = \text{BUS}(\text{WRITE}_\pi(c_2))$ for any two chips $c_1, c_2 \in C$ then we have $\phi_i(c_1) = \phi_i(c_2)$, which implies that $c_1 = c_2$, since ϕ_i is invertible. Conditions 4 and 5 both hold since $\text{LABEL}(\text{WRITE}_\pi(c)) = i$ and $\text{LABEL}(\text{READ}_\pi(c)) = j$ for all chips $c \in C$. We therefore conclude that the architecture $\mathcal{A}$ uniformly realizes Π. The architecture is uniform, but Theorem 2 obviates the need to show this fact. ■

Given a difference cover of small cardinality, Theorem 4 says we can construct a uniform architecture with few pins per chip. In fact, the reverse is true as well, as the following theorem shows.

Theorem 5 *Let Π be a permutation set, and let $\mathcal{A}$ be a uniform architecture for Π with k pins per chip. Then Π has a difference cover Φ such that $|\Phi| \leq k$.*

Proof. Given a uniform architecture $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL}\rangle$ for the permutation set Π, where k is the number of pins on each chip, we construct a difference cover Φ for Π as follows. Assume without loss of generality that $C = B = [n]$ and $\text{range}(\text{LABEL}) = [k]$. For each pin number i, where $i = 0, \ldots, k-1$, we define ϕ_i by $\phi_i(c) = b$ if and only if chip c is connected via its pin number i to bus b. We now define the difference cover Φ to be the set $\Phi = \{\phi_0, \phi_1, \ldots, \phi_{k-1}\}$. (The set Φ may have less than k elements, since some permutations may be repeated among the ϕ_i's.)

To see that Φ is a difference cover for Π, consider any permutation $\pi \in \Pi$. Since $\mathcal{A}$ uniformly realizes π, there exists a pair of pin labels (i, j) such that π is realized by each chip writing to its pin numbered i and reading from its pin numbered j. The labels i and j satisfy $i = \text{LABEL}(\text{WRITE}_\pi(c))$ and $j = \text{LABEL}(\text{READ}_\pi(c))$ for an all chips $c \in C$, as follows from Conditions 4 and 5 of Definition 3. Conditions 1 and 3 of Definition 3 imply that ϕ_i and ϕ_j are both permutations, and therefore there are $\phi_h, \phi_l \in \Phi$ such that $\phi_h = \phi_i$ and $\phi_l = \phi_j$. Finally, Condition 2 of Definition 3 implies that $\pi = \phi_j^{-1}\phi_i = \phi_l^{-1}\phi_h$, which proves that Φ is indeed a difference cover for Π. ■

Theorems 4 and 5 show that uniform architectures and difference covers are very closely related. Thus, when designing a uniform permutation architecture for a set of permutations, it suffices to focus on the problem of constructing a good difference cover for that set.

The structure of a permutation set can be helpful in obtaining a difference cover for it. In Sections 4 and 5, we investigate the construction of difference covers for cyclic groups of permutations and for groups in general. Here, we examine permutation sets formed by Cartesian products.

Definition 6 Let Π_1 be a set of permutations from X_1 to X_1, and let Π_2 be a set of permutations from X_2 to X_2. The *Cartesian product* $\Pi = \Pi_1 \times \Pi_2$ is the set of permutations from $X_1 \times X_2$ to $X_1 \times X_2$ defined as $\Pi = \{(\pi_1, \pi_2) : \pi_1 \in \Pi_1, \pi_2 \in \Pi_2\}$. Operations on the elements of Π are performed componentwise.

The Cartesian product $\Pi_1 \times \Pi_2$ is isomorphic to the Cartesian product $\Pi_2 \times \Pi_1$. The Cartesian product $\Pi = \Pi_1 \times \Pi_2$ is an abelian permutation set if and only if both Π_1 and Π_2 are abelian permutation sets.

The next two lemmas provide bounds on the size of difference covers for Cartesian products of permutation sets. (Similar lemmas hold for composition products of permutation sets.)

Lemma 6 *Let Π_1 be a permutation set on n_1 objects, and let Π_2 be a permutation set on n_2 objects. Then the Cartesian product $\Pi = \Pi_1 \times \Pi_2$, which is a permutation set on $n_1 \cdot n_2$ objects, has a difference cover of size $|\Pi_1| + |\Pi_2|$.*

Proof. Let Φ be the union of $\left\{(\pi_1^{-1}, I_{n_2}) : \pi_1 \in \Pi_1\right\}$ and $\{(I_{n_1}, \pi_2) : \pi_2 \in \Pi_2\}$. Each permutation $\pi = (\pi_1, \pi_2) \in \Pi$, can be represented as $(\pi_1, \pi_2) = (\pi_1^{-1}, I_{n_2})^{-1} \cdot (I_{n_1}, \pi_2)$, where both (π_1^{-1}, I_{n_2}) and (I_{n_1}, π_2) are in Φ. Thus Φ is a difference cover for Π, and the size of Φ is exactly $|\Pi_1| + |\Pi_2|$. ■

Lemma 7 *Let Π_1 be a permutation set on n_1 objects with a difference cover Φ_1, and let Π_2 be a permutation set on n_2 objects with a difference cover Φ_2. Then the Cartesian product $\Phi = \Phi_1 \times \Phi_2$ is a difference cover for $\Pi = \Pi_1 \times \Pi_2$.*

Proof. For each $\pi = (\pi_1, \pi_2) \in \Pi$, there exist $\phi_{i_1}, \phi_{j_1} \in \Phi_1$ such that $\pi_1 = \phi_{j_1}^{-1}\phi_{i_1}$, and there exist $\phi_{i_2}, \phi_{j_2} \in \Phi_2$ such that $\pi_2 = \phi_{j_2}^{-1}\phi_{i_2}$. We then have $(\pi_1, \pi_2) = (\phi_{j_1}^{-1}\phi_{i_1}, \phi_{j_2}^{-1}\phi_{i_2}) = (\phi_{j_1}, \phi_{j_2})^{-1}(\phi_{i_1}, \phi_{i_2})$, where both (ϕ_{i_1}, ϕ_{i_2}) and (ϕ_{j_1}, ϕ_{j_2}) are in $\Phi = \Phi_1 \times \Phi_2$, and hence Φ is a difference cover for Π. ■

To demonstrate both the use of difference covers and Lemma 7, we present in Figure 2 a uniform permutation architecture due to C. Fiduccia [10] for realizing shifts in a two-dimensional array. The architecture uniformly realizes the permutation set $\Pi = \{\mathrm{I}, \mathrm{N}, \mathrm{E}, \mathrm{S}, \mathrm{W}, \mathrm{NE}, \mathrm{SE}, \mathrm{NW}, \mathrm{SW}\}$ of eight compass directions plus the identity I. We introduce two permutation sets $\Pi_1 = \{\mathrm{I}, \mathrm{N}, \mathrm{S}\}$, $\Pi_2 = \{\mathrm{I}, \mathrm{E}, \mathrm{W}\}$, and corresponding difference covers $\Phi_1 = \{\mathrm{I}, \mathrm{S}\}$ and $\Phi_2 = \{\mathrm{I}, \mathrm{E}\}$. The Cartesian product $\Pi_1 \times \Pi_2$ is Π, and the set of permutations $\Phi = \Phi_1 \times \Phi_2 = \{\mathrm{S}, \mathrm{SE}, \mathrm{E}, \mathrm{I}\}$ is a difference cover for Π.

We conclude this section by defining the notion of a substring cover for a permutation set Π, which is equivalent to the notion of a difference cover. (A similar notion for difference sets is well known in the literature [6, 26].)

Definition 7 An ordered list $\Sigma = \langle \sigma_0, \sigma_1, \ldots, \sigma_{k-1} \rangle$ of permutations is a *substring cover* for a permutation set Π if

1. $\sigma_0\sigma_1 \cdots \sigma_{k-1} = I$, and
2. for all $\pi \in \Pi$, there exist $0 \leq i, j \leq k-1$ such that $\pi = \sigma_i\sigma_{i+1} \cdots \sigma_j$, where the arithmetic in the indices is performed modulo k.

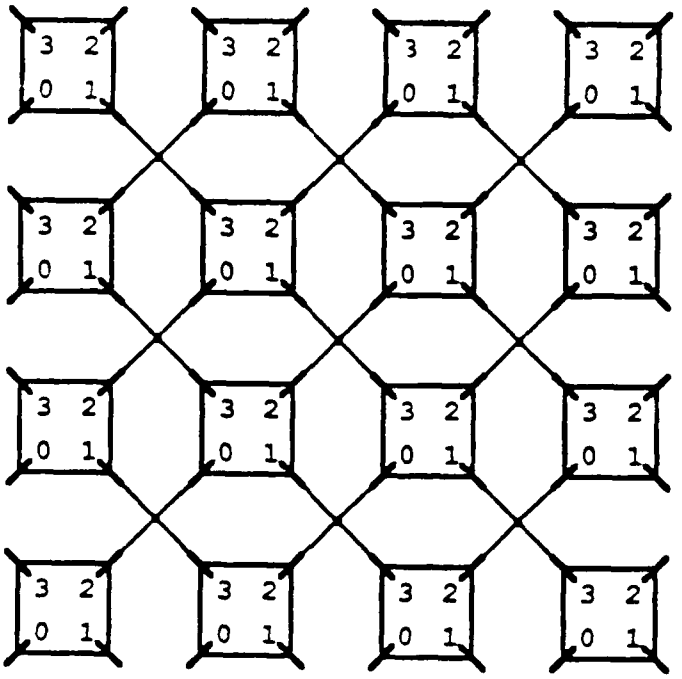


Figure 2: A uniform architecture due to C. Fiduccia [10] based on the difference cover {S, SE, E, I} for the permutation set $\Pi = \{I, N, E, S, W, NE, SE, NW, SW\}$.

The substring cover Σ is a list of permutations such that all the permutations in Π can be represented as a composition of a substring of permutations of Σ. The following two theorems show that the notions of a substring cover and difference cover are equivalent.

Theorem 8 *Let Π be a permutation set on n elements, and let Σ be a k-element substring cover for Π. Then Π has a difference cover Φ with at most k elements.*

Proof. Given a k-element substring cover $\Sigma = \langle \sigma_0, \sigma_1, \ldots, \sigma_{k-1} \rangle$ for Π, a difference cover Φ with at most k elements can be constructed. For each $0 \leq i \leq k-1$ we define $\phi_i = \sigma_0 \sigma_1 \cdots \sigma_i$. If a permutation π can be represented as $\pi = \sigma_i \sigma_{i+1} \cdots \sigma_j$, then $\pi = \phi_{i-1}^{-1} \phi_j$. By construction, the difference cover Φ has at most k elements. ∎

Theorem 9 *Let Π be a permutation set on n elements, and let Φ be a k-element difference cover for Π. Then Π has a substring cover Σ with k elements.*

Proof. Given a k-element difference cover $\Phi = \{\phi_0, \phi_1, \ldots, \phi_{k-1}\}$ for Π, we build a substring cover Σ for Π by defining $\sigma_i = \phi_{i-1}^{-1} \phi_i$ for all $0 \leq i \leq k-1$. The product $\sigma_0 \cdots \sigma_{k-1}$ yields the identity permutation. For each $\pi \in \Pi$, if $\pi = \phi_i^{-1} \phi_j$, then $\pi = \sigma_{i+1} \sigma_{i+2} \cdots \sigma_j$. Therefore Σ is a substring cover for Π with k elements. ∎

Referring back to the example of the eight compass directions, we present a substring cover for the permutation set $\Pi = \{I, N, E, S, W, NE, SE, NW, SW\}$. The substring cover

$\Sigma = \langle S, E, N, W \rangle$ is constructed from the difference cover $\Phi = \{S, SE, E, I\}$ that was used in the architecture of Figure 2. Each of the eight compass directions can be realized as a substring of the list $\Sigma = \langle S, E, N, W \rangle$.

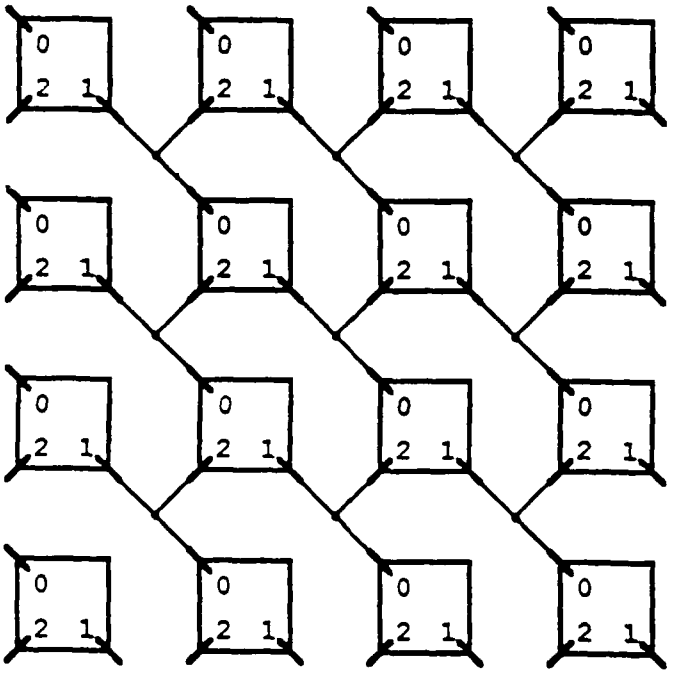


Figure 3: A uniform architecture due to C. Feynman [15] based on the difference cover $\{N, E, I\}$ for the permutations set $\Pi = \{I, N, E, S, W\}$.

As another example, consider the permutation set $\Pi = \{I, N, E, S, W\}$ of the shifts in a 2-dimensional array corresponding to the four compass directions. This permutation set has a difference cover $\Phi = \{N, E, I\}$ and a corresponding substring cover $\Sigma = \langle N, SE, W \rangle$. Consequently, there is a uniform architecture for realizing the four compass directions with three pins per chip, as has been observed by C. Feynman [15, pp. 437–438]. Figure 3 presents a uniform architecture based on the difference cover $\Phi = \{N, E, I\}$ for the permutation set $\Pi = \{I, N, E, S, W\}$.

4 Cyclic shifters

This section describes uniform architectures for realizing cyclic shifts among n chips in one clock tick. We first present a difference cover of size $O(\sqrt{n})$ for the set of all n cyclic shifts on n elements, and we give an area-efficient layout for the corresponding permutation architecture suitable for implementation as a printed-circuit board. When n can be expressed as $n = q^2 + q + 1$, where q is a power of a prime, we improve the bound on the size of a difference cover for all cyclic shifts on n elements to the optimal value of

$\lceil\sqrt{n}\rceil$. Finally, we prove that for any cyclic shifter that operates in one clock tick (even a nonuniform one), the average number of pins per chip is at least $\lceil\sqrt{n}\rceil$.

The first permutation architecture for cyclic shifters that we present is based on the construction in the following simple theorem.

Theorem 10 *The set of n cyclic shifts on n elements has a difference cover of size at most $2\lceil\sqrt{n}\rceil - 1$.*

Proof. Since the set of n cyclic shifts on n elements forms a group, and since this group is isomorphic to the group $\mathbf{Z}_n$, we shall construct a difference cover D for $\mathbf{Z}_n$. For convenience, let $m = \lceil\sqrt{n}\rceil$. Define two sets $A = \{0, 1, \ldots, m-1\}$ and $B = \{0, m, 2m, \ldots, (m-1)m\}$, and let the difference cover D be defined by $D = A \cup B$. Each element $s \in \mathbf{Z}_n$ can be realized as $s = b - a \pmod{n}$, where $a \in A$ and $b \in B$ by taking $a = m - (s \bmod m)$ and $b = \lceil s/m \rceil \cdot m$, as can be verified. The size of the difference cover D is $2m - 1 = 2\lceil\sqrt{n}\rceil - 1$, since the element 0 occurs in both A and B. ■

The difference cover constructed in the proof of Theorem 10 corresponds to an architecture with a regular, area-efficient layout, as shown in Figure 4. The n chips of the architecture are laid out in an array consisting of $m = \sqrt{n}$ rows, each containing $\sqrt{n}$ chips. (For simplicity, we assume that n is a square.) Each chip has pins $0, 1, \ldots, m-1$ on the top side, and pins $m, m+1, \ldots, 2m-1$ on the left side. Each bus consists of one vertical segment and one or two horizontal segments. Each wiring channel consists of $m = \sqrt{n}$ tracks, where each track is used to lay out segments of busses. When n is not a square, a cyclic shifter on n chips can be laid out in a similar fashion, with each wiring channel having at most $2\lceil\sqrt{n}\rceil$ tracks. The side of the layout is therefore $O(n)$, since there are $\lceil\sqrt{n}\rceil$ chips and $\lceil\sqrt{n}\rceil$ wiring channels along the side. The area of the layout is $O(n^2)$, which is asymptotically optimal since any architecture that can realize any of the cyclic-shift permutations in one clock tick requires area $\Omega(n^2)$ [30, p. 56].

Remark. The bound of $2\lceil\sqrt{n}\rceil - 1$ pins per chip can be improved to $(\sqrt{2} + o(1))\sqrt{n}$. See Section 8.

Occasionally, it is desirable to implement a subset of the cyclic shifts on n elements. The following corollary to Theorem 10 shows that when the shift amounts form an arithmetic sequence, a small difference cover exists.

Corollary 11 *Let a, b, and p be integers modulo n. For each $r \in [p]$, define π_r to be the permutation on $[n]$ that maps each $c \in [n]$ to $c + a + rb \pmod{n}$. Then the permutation set $\{\pi_r : r \in [p]\}$ has a difference cover of size $2\lceil\sqrt{p}\rceil$.*

Proof. As in the proof of Theorem 10, we construct two sets A and B whose union is the desired difference cover. The sets are $A = \{0, b, 2b, \ldots, (m-1)b\}$ and $B = \{a, a+mb, a+2mb, \ldots, a+(m-1)mb\}$, where $m = \lceil\sqrt{p}\rceil$. ■

Returning to the problem of implementing all n cyclic shifts on n elements, the following theorem demonstrates that for certain values of n, the optimal $\lceil\sqrt{n}\rceil$ bound can be obtained.

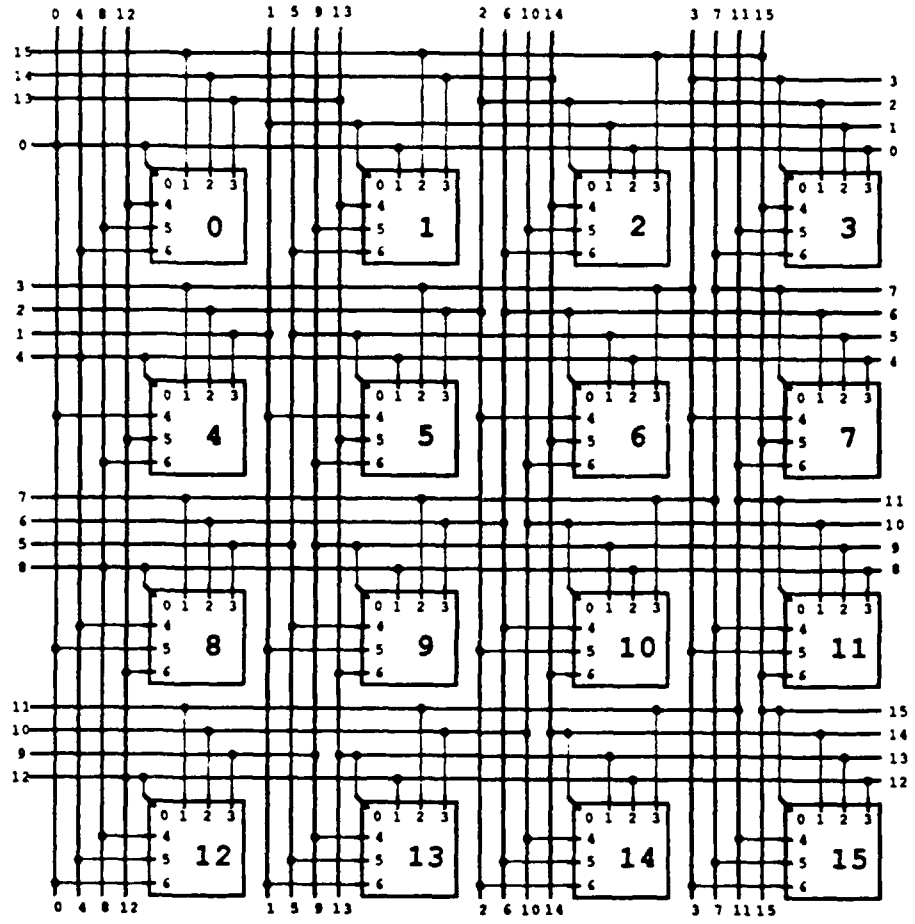


Figure 4: A layout for a cyclic shifter with $n = 16$ chips. Each chip and each bus has 7 pins. Each bus is constructed of one vertical segment and either one or two horizontal segments.

Theorem 12 *The set of n cyclic shifts on n elements has a difference cover of size $\lceil \sqrt{n} \rceil$ if $n = q^2 + q + 1$, where q is a power of a prime.*

Proof. As in the proof of Theorem 10, the problem is equivalent to that of constructing a difference cover D for $\mathbf{Z}_n$. When n is the size of a projective plane ($n = q^2 + q + 1$, where q is a power of a prime), this problem is equivalent to the problem of constructing a difference set. The difference set we give is due to Singer; a proof of its correctness is given in Hall [18, p. 129]. Let x be a primitive root of the Galois field $\mathrm{GF}(q^3)$, and let $F(y)$ be any irreducible cubic polynomial over the Galois field $\mathrm{GF}(q)$. We construct a difference cover D for $\mathbf{Z}_n$ from the set $[n]$ by choosing those $i \in [n]$ such that the power x^i can be written in the form $x^i = ax + b \pmod{F(x)}$ for some $a, b \in \mathrm{GF}(q)$. ∎

The construction of a uniform architecture based on a projective plane can be interpreted as follows. The n points of the projective plane correspond to the n chips and the n lines of the projective plane correspond to the n busses. Each line contains $q + 1$ points, which means that each bus is connected to $q + 1$ chips. Each point is incident on $q + 1$ lines, which means that each chip is connected to $q + 1$ different busses through its $q + 1$ pins. For example, Figure 1 demonstrates a uniform architecture based on the projective plane of size 13.

Theorems similar to Theorem 10 (but without application to architecture) appear in the combinatorics literature: see, for example, [22]. Bus connection networks based on projective planes have also been studied by Bermond, Bond, and Scalé [4] and by Mickunas [25], who observed that projective planes can be used to construct hypergraphs of diameter one.

Uniform architectures for cyclic shifters based on projective planes achieve the minimal number of pins per chip among all uniform cyclic shifters. We now prove a lower bound of $\lceil \sqrt{n} \rceil$ on the average number of pins per chip for any permutation architecture that realizes all the cyclic shifts. This lower bound applies to all permutation architectures, including nonuniform ones, and shows that uniform cyclic shifters based on projective planes are optimal among all cyclic shifters that operate in a single clock tick.

Theorem 13 *Let* $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL} \rangle$ *be a permutation architecture for the* n *cyclic shifts on* n *chips. Then the average number of pins per chip is at at least* $\lceil \sqrt{n} \rceil$.

Proof. The average number of pins per chip is $|P|/n$. We shall prove that $|P| \geq n \lceil \sqrt{n} \rceil$ which implies the theorem. We adopt the following conventions for notational convenience:

1. The set of busses is $B = \{b_0, b_1, \ldots, b_{m-1}\}$. We denote by k_i the number of pins connected to bus b_i, that is, $k_i = |\{p \in P : \text{BUS}(p) = b_i\}|$.

2. The busses that have at least $\lceil \sqrt{n} \rceil$ pins each are indexed first, that is, if there are r busses with at least $\lceil \sqrt{n} \rceil$ pins each, then $k_i \geq \lceil \sqrt{n} \rceil$ for $i = 0, \ldots, r-1$ and $k_i < \lceil \sqrt{n} \rceil$ for $i = r, \ldots, m-1$.

The thrust of the proof is to count the number of distinct data transfers when the architecture realizes each of the $n-1$ nontrivial shifts in turn. (The identity permutation is a trivial shift.) Each chip can be mapped to each other chip by one of the cyclic shifts, i.e., the cyclic shifts form a transitive group of permutations. Considering only the $n-1$ nontrivial shifts, there are exactly $n(n-1)$ distinct data transfers that must be implemented through interconnections in the architecture.

We compute an upper bound on the number of distinct data transfers that the busses can implement. Each of the first r busses $b_0, \ldots, b_{r-1}$ can be employed to realize at most one distinct data transfer in each of the $n-1$ nontrivial shifts. Thus, at most $r(n-1)$ distinct data transfers can be carried out by the first r busses. Any other bus b_i, where $r \leq i \leq m-1$, can realize at most $k_i(k_i-1)$ distinct nontrivial data transfers, since it has only k_i pins connected to it. Thus, the total number of distinct data transfers that the busses can realize is

$$r(n-1) + \sum_{i=r}^{m-1} k_i(k_i-1) \ ,$$

which must be larger than $n(n-1)$ if all nontrivial shifts are to be realized. Hence, we have

$$\sum_{i=r}^{m-1} k_i(k_i-1) \geq (n-r)(n-1) \ .$$

We can use this inequality to bound the number of pins on all busses with fewer than $\lceil\sqrt{n}\rceil$ pins. We have $k_i - 1 \leq \lceil\sqrt{n}\rceil - 2$ for $i = r, \ldots, m-1$, and thus

$$\begin{aligned}
\sum_{i=r}^{m-1} k_i &\geq \frac{1}{\lceil\sqrt{n}\rceil - 2} \sum_{i=r}^{m-1} k_i(k_i - 1) \\
&\geq \frac{(n-r)(n-1)}{\lceil\sqrt{n}\rceil - 2} \\
&\geq (n-r)\left\lceil\sqrt{n}\right\rceil .
\end{aligned}$$

We now bound the total number of pins in the architecture from below. We have

$$\begin{aligned}
|P| &= \sum_{i=0}^{m-1} k_i \\
&= \sum_{i=0}^{r-1} k_i + \sum_{i=r}^{m-1} k_i \\
&\geq r\left\lceil\sqrt{n}\right\rceil + (n-r)\left\lceil\sqrt{n}\right\rceil \\
&= n\left\lceil\sqrt{n}\right\rceil ,
\end{aligned}$$

which proves the theorem. ■

5 Difference covers for groups

In this section we show that small difference covers for abelian and nonabelian permutation groups exist. Specifically, for any permutation group Π with p elements, we show how to construct a difference cover with $O(\sqrt{p \lg p})$ elements. In the case where Π is abelian, we apply the decomposition theorem for finite abelian groups and the results for cyclic shifters in Section 4 to sharpen this bound to $O(\sqrt{p})$, which is optimal to within a constant factor.

As the first result of this section, we give a method for constructing a small difference cover for an arbitrary permutation group.

Theorem 14 *Let Π be an arbitrary group with p elements. Then Π has a difference cover Φ of size at most $\sqrt{2p \ln p} + 1$.*

Proof. We construct a difference cover incrementally starting with a partial difference cover $\Phi_1 = \{I\}$. At each step of the construction, we select an element $\phi_{i+1} \in \Pi$ such that $\left|\Phi_i^{-1}(\Phi_i \cup \{\phi_{i+1}\})\right|$ maximizes $\left|\Phi_i^{-1}(\Phi_i \cup \{\pi\})\right|$ over all $\pi \in \Pi$. We then define the new partial difference cover as $\Phi_{i+1} = \Phi_i \cup \{\phi_{i+1}\}$.

The analysis of this construction is in three parts. We first determine a lower bound on the number of elements of Π that are not covered by the partial difference cover Φ_i but are covered by Φ_{i+1}. We then develop a recurrence to upper bound the number of elements

of the group Π that are not covered at the ith step. Finally, we solve the recurrence to determine that the number k of iterations needed to cover all elements in Π is at most $\sqrt{2p \ln p} + 1$.

We first determine how many new elements of Π are covered when Φ_i is augmented with ϕ_{i+1} to produce Φ_{i+1}, for $i \geq 1$. Let the set Δ_i be the set of elements that are not covered by the partial difference cover Φ_i, which can be defined as $\Delta_i = \Pi - \Phi_i^{-1}\Phi_i$. Consider triples of the form $\langle \phi, \delta, \pi \rangle$ such that $\phi \in \Phi_i$, $\delta \in \Delta_i$, $\pi \in \Pi$, and $\phi\delta = \pi$. Observe that for any fixed $\pi \in \Pi$ and $\delta \in \Delta_i$, there is at most one triple of the form $\langle \phi, \delta, \pi \rangle$ in the set of triples, namely $\langle \pi\delta^{-1}, \delta, \pi \rangle$ when $\pi\delta^{-1} \in \Phi_i$. For a fixed π, the number of triples $\langle \phi, \delta, \pi \rangle$ in the set of triples is a lower bound on the number of elements covered by $\Phi_i \cup \{\pi\}$ but not by Φ_i, since we have $\delta = \phi^{-1}\pi$ and $\delta \in \Delta_i = \Pi - \Phi_i^{-1}\Phi_i$. For each $\phi \in \Phi_i$ and $\delta \in \Delta_i$, there is exactly one triple in the set of triples, and thus there are exactly $|\Phi_i| \cdot |\Delta_i|$ triples. Since there are at most $|\Pi|$ distinct permutations appearing as the third coordinate of a triple, the permutation ϕ_{i+1} that appears most often must appear at least $|\Phi_i| \cdot |\Delta_i| / |\Pi|$ times, and hence at least this many elements are covered by Φ_{i+1} that are not covered by Φ_i.

We can now bound the number of elements not covered by Φ_{i+1} in terms of the number of elements not covered by Φ_i by

$$\begin{aligned} |\Delta_{i+1}| &\leq |\Delta_i| - \frac{|\Phi_i| \cdot |\Delta_i|}{|\Pi|} \\ &= |\Delta_i| \left(1 - \frac{i}{p}\right) \\ &= |\Delta_1| \prod_{j=1}^{i} \left(1 - \frac{j}{p}\right) \\ &< p \prod_{j=1}^{i} \left(1 - \frac{j}{p}\right). \end{aligned}$$

When we obtain $|\Delta_k| < 1$ for some k, the partial difference cover Φ_k is a difference cover for Π because Δ_k is empty. Thus, Φ_k is a difference cover when

$$p \prod_{j=1}^{k-1} \left(1 - \frac{j}{p}\right) \leq 1,$$

or equivalently, when

$$\ln p + \sum_{j=1}^{k-1} \ln \left(1 - \frac{j}{p}\right) \leq 0.$$

Using the inequality $\ln(1 + x) \leq x$, we have

$$\begin{aligned} \ln p + \sum_{j=1}^{k-1} \ln \left(1 - \frac{j}{p}\right) &\leq \ln p - \sum_{j=1}^{k-1} \frac{j}{p} \\ &= \ln p - \frac{1}{p} \sum_{j=1}^{k-1} j \end{aligned}$$

$$\begin{aligned} &\leq \ln p - \frac{(k-1)^2}{2p} \\ &\leq 0. \end{aligned}$$

Thus, Φ_k is a difference cover when $k \geq \sqrt{2p \ln p} + 1$. ■

This proof of Theorem 14 provides a construction which can be implemented as an deterministic, polynomial-time algorithm with $O(p^2 \lg p)$ algebraic steps. We could also have proved the theorem by relying on the result of Babai and Erdős [2] that any group has a small set of generators, but this method would have produced only an existential (nonconstructive) result.

We have shown that there are difference covers of size $O(\sqrt{p \lg p})$ for general permutation groups with p elements. We now show that if the group is abelian, difference covers of size $O(\sqrt{p})$ exist.

Theorem 15 *For any abelian group Π with p elements, there exists a difference cover Φ of size at most $3\sqrt{p}$.*

Proof. Assume without loss of generality that $p > 1$. By the decomposition theorem for finite abelian groups [23, p. 133], any abelian group Π is isomorphic to a cross product of cyclic groups

$$\Pi \approx \mathbf{Z}_{p_1} \times \mathbf{Z}_{p_2} \times \cdots \times \mathbf{Z}_{p_k},$$

where $p_1 p_2 \cdots p_k = p$, and each $p_j \geq 2$. Let i be the unique index such that $p_1 p_2 \cdots p_{i-1} \leq \sqrt{p}$ and $p_{i+1} p_{i+2} \cdots p_k < \sqrt{p}$, and let $m = \left\lceil \sqrt{p} / p_1 p_2 \cdots p_{i-1} \right\rceil$. Using the argument of Theorem 10, we first construct a difference cover for $\mathbf{Z}_{p_i}$ from the union of two sets A_i and B_i, where $|A_i| \leq m$ and $|B_i| \leq \lfloor p_i / m \rfloor$, such that each element of $\mathbf{Z}_{p_i}$ can be expressed in the form $b - a \pmod{p_i}$ or $a - b \pmod{p_i}$, where $a \in A_i$ and $b \in B_i$.

We now construct a difference cover for $\Pi \approx \mathbf{Z}_{p_1} \times \mathbf{Z}_{p_2} \times \cdots \times \mathbf{Z}_{p_k}$ from the union of two sets A and B, where

$$A \approx \mathbf{Z}_{p_1} \times \mathbf{Z}_{p_2} \times \cdots \times \mathbf{Z}_{p_{i-1}} \times A_i,$$

and

$$B \approx B_i \times \mathbf{Z}_{p_{i+1}} \times \mathbf{Z}_{p_{i+2}} \times \cdots \times \mathbf{Z}_{p_k}.$$

That $A \cup B$ is a difference cover for Π follows from essentially the same argument as is used in Lemma 7.

The size of the difference cover $A \cup B$ is $|A| + |B|$. The size of A is

$$\begin{aligned} |A| &= p_1 p_2 \cdots p_{i-1} |A_i| \\ &\leq p_1 p_2 \cdots p_{i-1} m \\ &\leq p_1 p_2 \cdots p_{i-1} \left\lceil \sqrt{p} / p_1 p_2 \cdots p_{i-1} \right\rceil \\ &\leq \sqrt{p} + p_1 p_2 \cdots p_{i-1} \\ &\leq 2\sqrt{p}. \end{aligned}$$

Similarly, the size of B is

$$\begin{aligned} |B| &= |B_i|\, p_{i+1}p_{i+2}\cdots p_k \\ &\le \lfloor p_i/m \rfloor\, p_{i+1}p_{i+2}\cdots p_k \\ &\le \left(p_i / \lceil \sqrt{p}\,/p_1p_2\cdots p_{i-1}\rceil\right) p_{i+1}p_{i+2}\cdots p_k \\ &\le \left(p_1p_2\cdots p_i/\sqrt{p}\,\right) p_{i+1}p_{i+2}\cdots p_k \\ &= \sqrt{p}\,. \end{aligned}$$

Consequently, the size of the difference cover for Π is at most $3\sqrt{p}$. ■

6 Multiple clock ticks

In this section we discuss uniform permutation architectures that realize permutations in several clock ticks. By using more than one clock tick, further savings in the number of pins per chip can be obtained. We generalize the notion of a difference cover to handle multiple clock ticks, and describe a cyclic shifter on n chips with only $O(n^{1/2t})$ pins per chip that operates in t ticks.

We first generalize the notion of a difference cover to handle realization of permutations in $t \ge 1$ clock ticks.

Definition 8 A *t-difference cover* for a permutation set Π is a set Φ of permutations such that $(\Phi^{-1}\Phi)^t \supseteq \Pi$.

Using a t-difference cover Φ for the permutation set Π, any permutation $\pi \in \Pi$ can be expressed as the composition of t differences of permutations from Φ. The next lemma relates t-difference covers to permutation architectures that realize permutations in t clock ticks.

Lemma 16 *Let Φ be a t-difference cover with k elements for a permutation set Π. Then there is a permutation architecture with k pins per chip that uniformly realizes Π in t clock ticks.*

Proof. We define the permutation set $\Sigma = \Phi^{-1}\Phi$. Let $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL}\rangle$ be the permutation architecture, based on the difference cover Φ, that uniformly realizes Σ. Hence, the permutation architecture $\mathcal{A}$ can uniformly realize any $\sigma \in \Sigma$ in one clock tick. Each permutation $\pi \in \Pi$ can be expressed as $\pi = \sigma_{t-1}\sigma_{t-2}\cdots\sigma_0$, where $\sigma_i \in \Sigma$ for $0 \le i \le t-1$, since we have $\Sigma^t = (\Phi^{-1}\Phi)^t \supseteq \Pi$. In order to realize π in t clock ticks, the permutation architecture $\mathcal{A}$ uniformly realizes σ_i in clock tick i for $0 \le i \le t-1$. ■

Lemma 16 claims that the problem of uniformly realizing a permutation set Π in t clock ticks can be reduced to finding a permutation set Σ such that $\Sigma^t \supseteq \Pi$, and then finding a difference cover for Σ. The great advantage of using more than one clock tick is in the further savings in the number of pins per chip. The following theorem, for example,

describes a construction of a t-difference cover of size $O(n^{1/2t})$ for the set of cyclic shifts on n objects. This result can be used to build a uniform architecture on n chips with only $O(n^{1/2t})$ pins per chip that can realize any cyclic shift on the n chips in t clock ticks.

Theorem 17 *For any $n \geq 1$ and $t \geq 1$, the permutation set of all the n cyclic shifts on n objects has a t-difference cover of size $O(n^{1/2t})$.*

Proof. For the purpose of the proof, we denote the permutation set of all the n cyclic shifts on n objects by Π_n. (We remind that $\Pi_n \approx \mathbf{Z}_n$.) We first treat the case for those n such that there exists an integer m satisfying $n^{1/t} \leq m \leq 4n^{1/t}$ and $\gcd(m,n) = 1$. We then use this case to extend the proof to all values of n.

Since $\gcd(m,n) = 1$, there exists an $m^{-1} \in \mathbf{Z}_n$ such that $m \cdot m^{-1} = 1 \pmod n$. For each $r \in [m]$, define the permutation $\sigma_r : [n] \to [n]$ as $\sigma_r(c) = m^{-1}(c + r) \pmod n$, and define the permutation $\sigma'_r : [n] \to [n]$ as $\sigma'_r(c) = m^{t-1}(c + r) \pmod n$. Next define the permutation set $\Sigma = \{\sigma_r\} \cup \{\sigma'_r\}$. The set $\{\sigma_r\}$ is an arithmetic sequence of cyclic shifts on n elements (as in Corollary 11) followed by the fixed permutation corresponding to multiplication by m^{-1}, and thus $\{\sigma_r\}$ has a difference cover of size $O(\sqrt{m})$. Similarly, the set $\{\sigma'_r\}$ has a difference cover of size $O(\sqrt{m})$. Combining the two difference covers for $\{\sigma_r\}$ and $\{\sigma'_r\}$, we get a difference cover Φ of size $O(\sqrt{m}) = O(n^{1/2t})$ for Σ.

We now show the inclusion $\Sigma^t \supseteq \Pi_n$. Let $\pi \in \Pi_n$ be a permutation of a cyclic shift by s. We express the shift amount $s \in [n]$ as $s = s_0 + s_1 m + \cdots + s_{t-1} m^{t-1}$, where $s_i \in [m]$ for $0 \leq i \leq t-1$. The permutation π can be described as

$$\begin{aligned}
\pi(c) &= c + s \pmod n \\
&= c + s_0 + s_1 m + \cdots + s_{t-1} m^{t-1} \pmod n \\
&= m^{t-1}\left(s_{t-1} + m^{-1}\left(s_{t-2} + \cdots + m^{-1}\left(s_0 + c\right)\right)\right) \pmod n \\
&= \sigma'_{s_{t-1}} \sigma_{s_{t-2}} \cdots \sigma_{s_0}(c),
\end{aligned}$$

which proves that $\pi \in \Sigma^t$. Hence, we get the inclusion $\Sigma^t \supseteq \Pi_n$, which together with the fact that there is a difference cover Φ of size $O(n^{1/2t})$ for Σ, proves the theorem for the case when there exists an integer m satisfying $n^{1/t} \leq m \leq 4n^{1/t}$ and $\gcd(m,n) = 1$.

Such an m need not exist for every n and every t, however. We can overcome this difficulty by factoring $n = n_1 n_2$ such that n_1 consists of no even-indexed primes (3, 7, 13, ...) and n_2 consists of no odd-indexed primes (2, 5, 11, ...). Since we have $\gcd(n_1, n_2) = 1$, we can use the Chinese remainders theorem to express $\mathbf{Z}_n$ as a Cartesian product $\mathbf{Z}_n \approx \mathbf{Z}_{n_1} \times \mathbf{Z}_{n_2}$. We let m_1 be the first even-indexed prime at least as large as $n_1^{1/t}$, and let m_2 be the first odd-indexed prime at least as large as $n_2^{1/t}$. Bertrand's postulate [19, p. 343] guarantees that for every x, there is a prime between x and $2x$, which means $m_j \in [n_j^{1/t}, 4n_j^{1/t}]$ for $j = 1, 2$. (Tighter bounds are possible.)

We can now use the previous construction to construct a t-difference cover Φ_1 of size $O(n_1^{1/2t})$ for $\mathbf{Z}_{n_1}$, which is isomorphic to Π_{n_1}, and a t-difference cover Φ_2 of size $O(n_2^{1/2t})$ for $\mathbf{Z}_{n_2}$, which is isomorphic to Π_{n_2}. Using the same technique as in the proof of Lemma 7.

we can construct a t-difference cover of size $O(n_1^{1/2t}) \cdot O(n_2^{1/2t}) = O(n^{1/2t})$ for $\mathbf{Z}_{n_1} \times \mathbf{Z}_{n_2} \approx \mathbf{Z}_n \approx \Pi_n$. ■

One can rather straightforwardly use Corollary 11 to obtain a t-difference cover of size $O(tn^{1/2t})$. Based on the representation of the shift amount $s = s_0 + s_1 m + \cdots + s_{t-1} m^{t-1}$, one can come with t separate difference covers, each of size $O(n^{1/2t})$, for the t separate sequences of arithmetic shifts by $\{sm^i : s \in [m]\}$ for $0 \leq i \leq t-1$. Theorem 17 avoids the extra factor of t by constructing only one such difference cover and using its elements for each one of the t differences.

7 Extensions

This section contains some additional results on permutation architectures and difference covers. We describe efficient, uniform architectures that can realize the permutations implemented by various popular interconnection networks, including multidimensional meshes, hypercubes, and shuffle-exchange networks. We examine nonuniform permutation architectures, and adapt some combinatorial results in the literature to apply to permutation architectures. A result of DeBruijn leads to a nonuniform architecture with $O(\sqrt{n \lg n})$ pins per chip that can realize all $n!$ permutations on n chips.

7.1 Specific networks

By using busses, many popular interconnection networks can be realized with fewer pins than conventionally proposed. Here, we mention a few.

The permutation architectures for realizing compass shifts on two-dimensional arrays can be extended in a natural fashion to d-dimensional arrays. For the d-dimensional analogue of the shifts $\{\mathrm{I}, \mathrm{N}, \mathrm{E}, \mathrm{S}, \mathrm{W}\}$, there is a uniform architecture that uses only $d+1$ pins per chip to implement the $2d+1$ permutations. For the d-dimensional analogue of the shifts $\{\mathrm{I}, \mathrm{N}, \mathrm{E}, \mathrm{S}, \mathrm{W}, \mathrm{NE}, \mathrm{SE}, \mathrm{NW}, \mathrm{SW}\}$, there is a uniform architecture that uses only 2^d pins per chip to implement the 3^d permutations. (These two results were independently discovered by C. Fiduccia [11, 12].)

A Boolean hypercube of dimension d is a degenerate case of a d-dimensional array. Only $d+1$ pins per chip are required by a permutation architecture that uses busses, whereas $2d$ pins per chip are needed if point-to-point wires are used. (To realize a swap of information across a dimension in one clock tick, each chip requires two pins for that dimension: one to read and one to write.)

A permutation architecture that implements the permutations Shuffle, Inverse Shuffle, and Exchange can be constructed with three pins per chip instead of the usual four, and it can implement the Shuffle-Exchange and Inverse Shuffle-Exchange permutations in one tick as well.

7.2 Average number of pins per chip

Theorem 13 presents a lower bound on the average number of pins per chip in any cyclic shifter that operates in one clock tick. The following theorem is a natural extension of Theorem 13 for a general set of permutations.

Theorem 18 *Let Π be a permutation set on n objects with p permutations and with total of T nontrivial data transfers, and let $\mathcal{A} = \langle C, B, P, \text{CHIP}, \text{BUS}, \text{LABEL} \rangle$ be any permutation architecture for realizing Π. Then the average number of pins per chip is at least $T/n\sqrt{p}$.*

Proof. As in the proof of Theorem 13, we prove that $|P| \geq T/\sqrt{p}$ which implies the theorem. We make similar notational conventions:

1. The set of busses is $B = \{b_0, b_1, \ldots, b_{m-1}\}$. We denote by k_i the number of pins connected to bus b_i.
2. The r busses that have at least $\sqrt{p}$ pins each are indexed first, that is $k_i \geq \sqrt{p}$ for $i = 0, \ldots, r-1$ and $k_i < \sqrt{p}$ for $i = r, \ldots, m-1$.

We count the number of distinct data transfers that can be accomplished by each bus. Each of the first r busses can be employed to realize at most p out of the T nontrivial data transfers, since it can be used at most once for each of the p permutation. Any other bus b_i, where $r \leq i \leq m-1$, can realize at most $k_i(k_i - 1)$ out of the T nontrivial data transfers, since it has only k_i pins connected to it. We need to have $\sum_{i=r}^{m-1} k_i(k_i - 1) \geq T - rp$, which implies

$$\begin{aligned} \sum_{i=r}^{m-1} k_i &\geq \frac{T - rp}{\sqrt{p}} \\ &= \frac{T}{\sqrt{p}} - r\sqrt{p}. \end{aligned}$$

The number of pins in the architecture can now be bounded as follows:

$$\begin{aligned} |P| &= \sum_{i=0}^{m-1} k_i \\ &= \sum_{i=0}^{r-1} k_i + \sum_{i=r}^{m-1} k_i \\ &\geq r\sqrt{p} + \left(\frac{T}{\sqrt{p}} - r\sqrt{p} \right) \\ &= \frac{T}{\sqrt{p}} \end{aligned}$$

■

Theorem 18 demonstrates that uniform architectures can achieve the optimal number (to within a constant factor) of pins per chip for certain classes of permutation sets. When there are relatively few permutations that are responsible for many nontrivial data transfers, the average number of pins per chip is high. The set of cyclic shifts is an example of this kind of permutation set.

7.3 Nonuniform architectures

When the uniformity condition on permutation architectures is dropped, one can do much better in terms of the number of pins per chip. The complexity of control may increase substantially, however, due to the irregular communication patterns and the number of possible permutations realizable for some of the architectures. Nevertheless, from a mathematical point of view, nonuniform architectures are quite interesting.

In fact, nonuniform architectures have been studied quite extensively in the mathematics literature in the guise of partitioning problems. For the problem of realizing all $n!$ permutations on n chips, a result due to de Bruijn, Erdös, and Spencer [31, p. 106-108] implies that $O(\sqrt{n \lg n})$ pins per chip suffice. The nonuniform architecture that achieves this bound is constructed probabilistically, however. It is an open problem to obtain this bound deterministically. The best deterministic construction to date is due to Feldman, Friedman, and Pippenger [9] and uses $O(n^{2/3})$ pins per chip.

8 Further research

In this section we list a few of the problems that have been left open by our research. We also describe briefly some further work brought on by an earlier version [20] of our work.

In Section 4 we described a difference cover of size $2\lceil\sqrt{n}\rceil - 1$ for the cyclic group $\mathbf{Z}_n$, and proved that when n is the order of a projective plane, there is a difference cover of size $\lceil\sqrt{n}\rceil$. It seems reasonable that any cyclic group $\mathbf{Z}_n$ might actually have a difference cover of size $\sqrt{n} + o(\sqrt{n})$, but we have been unable to prove or disprove this conjecture. Mills and Wiedemann [27] have computed a table of minimal difference covers for all the cyclic groups of cardinality up to 110. For any value of n up to 110, the difference cover they find has at most $\lceil\sqrt{n}\rceil + 2$ elements. They also provide [28] a "folk theorem" that establishes a stronger upper bound for the general case than $2\lceil\sqrt{n}\rceil - 1$.

Theorem 19 *The set of n cyclic shifts on n elements has a difference cover of size $(\sqrt{2} + o(1))\sqrt{n}$.*

Sketch of proof. [28] Let q be the smallest prime such that $l = q^2 + q + 1 \geq n/2$. We have $q = (1 + o(1))\sqrt{n/2}$, since for large x, there exists a prime between x and $x + o(x)$. Let $\{d_0, d_1, \ldots, d_q\}$ be a difference cover for *integers* l chosen as in Theorem 12. It can be verified that the set $\{d_0, d_1, \ldots, d_q\} \cup \{d_0 + l, d_1 + l, \ldots, d_q + l\}$ forms a difference cover for $\mathbf{Z}_n$. ∎

Another interesting problem related to cyclic shifters involves finding an area-efficient VLSI layout of the cyclic shifter based on projective planes. In section 4 we presented an area-efficient layout using a difference cover whose size is twice the optimal size. Is there a good layout for the pin-optimal design?

In Section 5, we showed that any abelian group of p elements has a difference cover of size $O(\sqrt{p})$, and we showed that any group of p elements has a difference cover of size

$O(\sqrt{p \lg p})$. Finkelstein, Kleitman and Leighton [13] have recently improved our result for general groups to $O(\sqrt{p})$. Their proof uses a folk theorem [8] that every simple group of nonprime order p has a subgroup of size at least $\sqrt{p}$. The folk theorem is proved by checking each type of group in the classification theorem [17, pp. 135–136]. It would be interesting to know if there is a more direct proof that every group has a difference cover of size $O(\sqrt{p})$.

To implement cyclic shifters that operate in t clock ticks, we showed how to construct a t-difference cover for $\mathbf{Z}_n$ of size $O(n^{1/2t})$. A simpler construction achieves the bound $O(tn^{1/2t})$. Theorem 13 gives a lower bound of $\lceil \sqrt{n} \rceil$ on the average number of pins per chip for a cyclic shifter that operates in one clock tick. It may be possible to prove a lower bound of $\Omega(n^{1/2t})$ on the average number of pins per chip when an architecture operates in t clock ticks, but we were unable to extend the argument. We were also unable to extend either of these constructions to give good t-difference covers for groups, either general or abelian. It would be interesting to know whether any abelian group of permutations with p permutations has a t-difference cover of size $O(tn^{1/2t})$, for any $t \geq 1$.

We have concentrated primarily on permutation sets that have good structure, specifically group properties. It would be interesting to identify other structural properties of permutation sets besides group properties that allow small difference covers to exist.

Appendix

For completeness, we include definitions of common mathematical notations and algebraic terms used in the paper. Definitions specific to the content of the paper are included in context.

We adopt the following notations:

- $|X|$ denotes the size of the set X.
- $[n]$ denotes the set of n integers $\{1, 2, \ldots, n\}$.
- $\lfloor x \rfloor$ (floor of x) denotes the largest integer that is smaller than or equal to x.
- $\lceil x \rceil$ (ceiling of x) denotes the smallest integer that is larger than or equal to x.
- $\lg x$ denotes $\log_2 x$.
- $\ln x$ denotes $\log_e x$.
- $\binom{n}{k}$ denotes $\frac{n!}{k!(n-k)!}$.

For two asymptotically positive functions $f(n)$ and $g(n)$, we write:

- $f(n) = o(g(n))$ if $\lim_{n \to \infty} f(n)/g(n) = 0$.
- $f(n) = O(g(n))$ if there exists $c > 0$ and n_0, such that $f(n) \leq cg(n)$ for all $n > n_0$.

- $f(n) = \Omega(g(n))$ if there exists $c > 0$ and n_0, such that $f(n) \geq cg(n)$ for all $n > n_0$.
- $f(n) = \Theta(g(n))$ if both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Let $f : A \rightarrow B$ be a function.

- f is *injective* (*one to one*) if $a \neq b$ implies $f(a) \neq f(b)$.
- f is *surjective* (*onto*) if for all $b \in B$, there exists some $a \in A$ such that $b = f(a)$.
- f is *bijective* if it is injective and surjective.

A *group* is a set of elements G with a binary operation $\oplus$, such that the following properties hold.

- *Closure:* For every $a, b \in G$, we have $a \oplus b \in G$.
- *Associativity:* For every $a, b, c \in G$, we have $a \oplus (b \oplus c) = (a \oplus b) \oplus c$.
- *Identity:* There exists an element $e \in G$ such that $a \oplus e = e \oplus a = a$ for all $a \in G$.
- *Inverse:* For every $a \in G$, there exists an element $a^{-1} \in G$ such that $a \oplus a^{-1} = a^{-1} \oplus a = e$.

An *abelian* group is a group G with an additional property:

- *Commutativity:* For every $a, b \in G$, we have $a \oplus b = b \oplus a$.

We often use the notations:

- ab to denote $a \oplus b$,
- a^k to denote $a \oplus a \oplus \cdots \oplus a$ (k times),
- a^{-k} to denote $(a^{-1})^k$.

A *cyclic* group G is a group in which there exists $a \in G$ such that $G = \left\{a^k : k \text{ integer}\right\}$. Cyclic groups are abelian. The notation $\mathbf{Z}_n$ denotes the cyclic group of residues modulo n, with modular addition as the group operation. A *permutation* on a set X is a bijective function from X to X. All the possible permutations on X form a group with functional composition as the group operation.

Acknowledgements

Guy L. Steele, Jr. of Thinking Machines Corporation originally acquainted us with the problem of implementing cyclic shifters with busses. Tom Leighton of MIT helped simplify our proof of Theorem 15 and acquainted us with references to relevant work in the combinatorics literature. Nicholas Pippenger of the University of British Columbia referred us to the combinatorics results in section 7.3. Noga Alon of Tel Aviv University made the observation in Section 5 that the result of Babai and Erdõs could be used to show the existence of a small difference cover for any group. Chuck Fiduccia of General Electric Research Center provided excellent comments and identified a few bugs in an early version of our paper. Dr. I. J. Matrix of the Massachusetts Institute of Theology acquainted us with his related work [14, pp. 65–67]. We thank these individuals for helpful discussions, as well as Benny Chor, Lance Fortnow, Shafi Goldwasser, Phil Klein, and Su-Ming Wu of MIT, and Andrew Odlyzko of AT&T Bell Laboratories. We would also like to thank the referees which provided excellent suggestions.

References

[1] A. Aggarwal, "Optimal bounds for finding maximum on array of processors with k global buses," *IEEE Transactions on Computers*, Vol. C-35, No. 1, January 1986, pp. 62–64.

[2] L. Babai and P. Erdõs, "Representation of group elements as short products," *Annals of Discrete Mathematics*, Vol. 12, 1982, pp. 27–30.

[3] J. C. Bermond, J. Bond, and C. Peyrat, "Interconnection network with each node on two buses," *Proceedings of the International Colloquium on Parallel Algorithms and Architectures*, Marseille Luminy, France, 1986, pp. 155–167.

[4] J. C. Bermond, J. Bond, and J. F. Scalé, "Large hypergraphs of diameter one," in *Graph Theory and Combinatorics, Proc. Coll. Cambridge, 1983*, Academic Press, London, 1984, pp. 19–28.

[5] Y. Birk, *Concurrent Communication among Multi-Transceiver Stations over Shared Media*, Ph.D. dissertation, Stanford University, March 1987.

[6] G. S. Bloom and S. W. Golomb, "Numbered complete graphs, unusual rulers, and assorted applications," in *Theory and Applications of Graphs*, Y. Alavi and D. R. Lick, eds., Springer-Verlag, New York, 1978.

[7] S. H. Bokhari, "Finding maximum on an array processor with a global bus," *IEEE Transactions on Computers*, Vol. C-33, No. 2, February 1984, pp. 133–139.

[8] W. Feit, private communications, 1987.

[9] P. Feldman, J. Friedman, and N. Pippenger, "Wide-sense nonblocking networks," *SIAM Journal of Discrete Mathematics,* Vol. 1, No. 2, May 1988, pp. 158–173.

[10] C. M. Fiduccia, public communication, MIT, 1984.

[11] C. M. Fiduccia, private communication, April 1987.

[12] C. M. Fiduccia, "A bussed hypercube and other optimal permutation networks," presented at the *4th SIAM Conference on Discrete Mathematics,* June 1988.

[13] L. Finkelstein, D. Kleitman, and T. Leighton, "Applying the classification theorem for finite simple groups to minimize pin count in uniform permutation architectures," in *VLSI Algorithms and Architectures,* Lecture Notes in Computer Science, Vol 319, J. H. Reif, ed., Springer-Verlag, New York, 1988, pp. 247–256.

[14] M. Gardner, *The Incredible Dr. Matrix,* Charles Scribner's Sons, New York, 1976.

[15] L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits,* Addison-Wesley, Reading, Massachusetts, 1985.

[16] S. W. Golomb, "How to number a graph," in *Graph Theory and Computing,* R. C. Read, ed., Academic Press, New York, 1972, pp. 23–37.

[17] D. Gorenstein, *Finite Simple Groups,* Plenum Press, New York, 1982.

[18] M. Hall, Jr., *Combinatorial Theory,* Blaisdell Publishing Company, Waltham, Massachusetts, 1967.

[19] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers,* Oxford University Press, London, 1938.

[20] J. Kilian, S. Kipnis, and C. E. Leiserson, "The organization of permutation architectures with bussed interconnections," *28th Annual Symposium on Foundations of Computer Science,* IEEE, October 12–14, 1987, pp. 305–315.

[21] T. Lang, M. Valero, and M. A. Fiol, "Reduction of connections for multibus organization," *IEEE Transactions on Computers,* Vol. C-32, No. 8, August 1983, pp. 707–715.

[22] J. Leech, "On the representation of $1, 2, \ldots, n$ by differences," *Journal of the London Mathematical Society,* Vol. 31, 1956, pp. 160–169.

[23] D. J. Lewis, *Introduction To Algebra,* Harper and Row, New York, 1965.

[24] R. J. Lipton and R. Sedgewick, "Lower bounds for VLSI," *13th Annual Symposium on Theory of Computing,* ACM, May 11–13, 1981, pp. 300–307.

[25] M. D. Mickunas, "Using projective geometry to design bus connection networks," *Proceedings of the Workshop on Interconnection Networks for Parallel and Distributed Processing,* ACM/IEEE, April 21–22, 1980, pp. 47–55.

[26] J. C. P. Miller, "Difference bases, three problems in additive number theory," in *Computers in Number Theory*, A. O. L. Atkin and B. J. Birch, eds., Academic Press, London, 1971, pp. 299–322.

[27] W. H. Mills and D. H. Wiedemann, "A table of difference coverings," unpublished abstract, Institute for Defense Analyses, Communications Research Division, January 1988.

[28] D. H. Wiedemann, private communication, November 1988.

[29] Q. F. Stout, "Meshes with multiple busses," *27th Annual Symposium on Foundations of Computer Science*, IEEE, October 27–29, 1986, pp. 264–273.

[30] J. D. Ullman, *Computational Aspects of VLSI*, Computer Science Press, Rockville, Maryland, 1984.

[31] J. H. van Lint, "Solutions: Problem 350," *Nieuw Archief voor Wiskunde*, Vol. 22, 1974, pp. 94–109.